DOCUMENT RESUME

ED 224 043 CS 207 285

AUTHOR Evans, Ron
TITLE The Friendly Interface for Computers and English Teachers.
PUB DATE Nov 82
NOTE 15p.; Paper presented at the Annual Meeting of the National Council of Teachers of English (72nd, Washington, DC, November 19-24, 1982).
PUB TYPE Guides - Classroom Use - Guides (For Teachers) (052) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Assisted Instruction; *Computer Literacy; *Computer Programs; *English Instruction; Higher Education; *Programing; *Programing Languages; Teacher Developed Materials

ABSTRACT

By exploring ways of designing computer-based instruction without having to learn a computer language and by learning ways of thinking that are compatible with the microcomputer, English teachers can take control of the computer in the classroom. Authoring languages are currently available that allow the teacher to input original subject material in personally styled language. Since computers also constrain thinking in very systematic ways, "Structured English," a limited set of complex, imperative sentences in which adverbial clauses set conditions or times for commands to be performed, has been developed for use with flow chart figures to create teaching programs. By following rhetorical guidelines developed through experience with "Structured English," teachers can create computer programs that actually engage or lead the learner with the intuitive insight of a live teacher. Sentences should be free of jargon and brief; commands should be invitational and supportive; choices should make a real difference; and the tone of all verb phrases should be friendly. Even with "Structured English," modified flow charts, and authoring systems, an English teacher still must be able to imagine what will be displayed on the video monitor each step of the sequence. A "script guide," a loose-leaf notebook providing precise communication between designer and programer, is an excellent tool for this purpose. (Figures include sample video displays of an authoring system.) (JL)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED224043

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

CS217110

THE FRIENDLY INTERFACE FOR COMPUTERS AND ENGLISH TEACHERS

Ron Evans
Professor of English
University of West Florida

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Ron Evans

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

CS207285

The Friendly Interface for Computers and English Teachers

Ron Evans

Professor of English, University of West Florida

No question about it, computers are here to stay in education. As the cost of operating schools escalates faster than the rate of inflation and as fewer students enter these schools, it's costing more to educate fewer. Thus the microcomputer, a one-time fixed-cost piece of hardware, becomes very attractive as a way to offset these astronomically rising costs. There is both optimism and omirousness in that prophecy, however.

The optimism evolves from the endless programming possibilities afforded by microcomputer, augmented with low-cost add-ons. The omniousness spreads like a dark cloud over teachers who fear replacement by this newest teaching machine. More than this, there are teachers who would like to avoid replacement by learning to use the computer, but programming seems too difficult to learn.

One way to resolve this problem is to explore ways of designing computer-based instruction without having to learn a computer language. Another is to learn ways of thinking that are compatible with the microcomputer, including the kinds of sentence structure best suited to computer displays. With these tools at their disposal, English teachers especially can resume the natural leadership and control of this latest communication and educational medium, in the same way they have done in the past with audiotape, film, and videotape throughout the 60's and 70's.

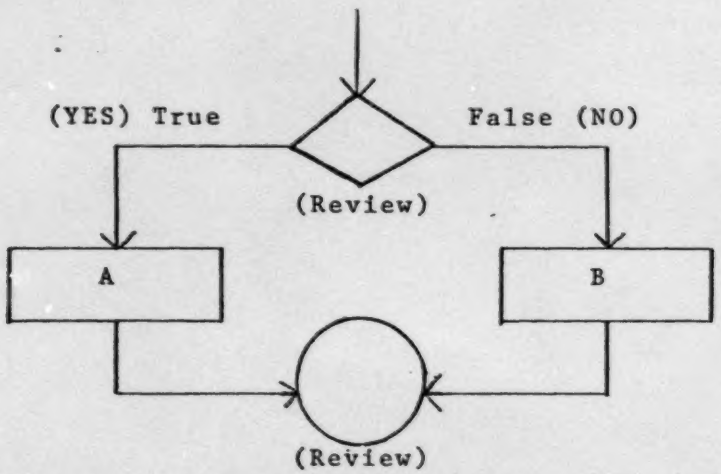
Among the useful aids in computer software are the so-called "authoring languages," such as "Apple Pilot," Super Pilot, and Bell and Howell's "PASS." These are computer programs, available on floppy diskettes, that allow the user to present directions or information, to phrase questions, and to predetermine allowable answers, all according to preestablished programmed sequences. In short, the authoring language acts like a directive outline in which the Roman numerals and some of the capital letters are filled in with conceptual statements like "Information Page," and "Question Page," and "How many questions do you want?" These provide, in effect, a menu through which users input original subject material in personally-styled language. Authoring languages also guide the users through sequenced steps to finish preparing an educationally sound lesson.

Computers also constrain thinking in very systematic ways. Learning to represent instruction in these systematic ways not only makes it possible for English teachers to move further in to such educational design, it also sharpens their awareness of disciplined communication strategies that enhance cognitive precision. One such system is called "Pseudocode," a form of which is referred to as "Structured English." Explalined by David Carew in the March '82 Microcomputing magazine, "Structured English" and "Pseudocode" work in tandem to develop computer programs "by a process involving successive levels of consideration and separation of concerns" (p. 54).

The separating of proposed teacher and learner concerns in any kind of written-down plan naturally must involve something

that looks like a flowchart. This apparent contradiction is resolved by Pseudocode and Structured English teaming within three basic box-like shapes and a like number of sentence structures to depict what will happen when in the computer-delivered plan. An example shows the normal appearance of a logical selection pattern, composed of a diamond, two rectangles, and a small circle, all connected by directional arrows.

FIGURE I



The diamond means one point of entry with three points of exit: either a "true-false" or "yes-no" judgment response, or a review of the original stimulus. The two rectangles, marked "A" and "B," are called "sequence boxes" and may house any number of statements, phrased in "Structured English" as follows:

IF (some conditional statement) THEN (some resulting statement)

ELSE (some other statement)

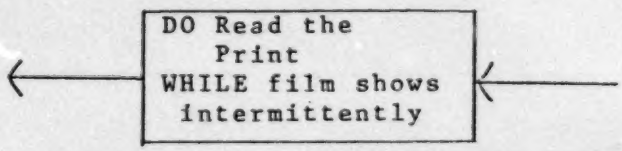
END IF (some other statement is fulfilled)

DO (directed behavior) WHILE (doing another directed behavior)

In effect, the so-called "Structured English" is nothing more than an extremely limited set of complex, imperative sentences, in which adverbial clauses set conditions or times for commands

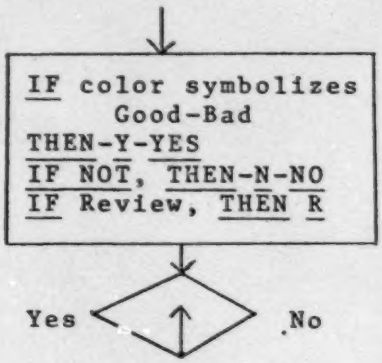
to be performed. Not unlike symbolic logic's skeltal propositions, these four program sentences constrain the planner's thinking to hypothetical conditions (IF-THEN), parallel alternates (ELSE), parallel correlaries (DO-WHILE), or conclusions (END IF). Beyond these, a variety of other sequenced guidelines shape the user's expression to include the presentation of information, the allowance for learner response entry, the simplified constraint of learner responses or review requests,and the branching of the learner to further information or review based on his or her response.

Naturally, with such a narrowed menu of interaction, "Structured English" could be stulitifying if not fleshed out in the most creative ways. There are few boundaries, for example, on what kind of stimulus is to be used--including motion and sound visuals from either video disc or video cassette. Consider, as an instance, that you wanted to display on the monitor a filmed excerpt for some reading passage such as Huckleberry Finn, to allow intermittent reading and viewing. This would be stated simply as a DO-WHILE command "Read the following print excerpts, while the screen alternately displays a film version of each passage," all of which would occur within one rectangular box, one arrow leading in, and one out.

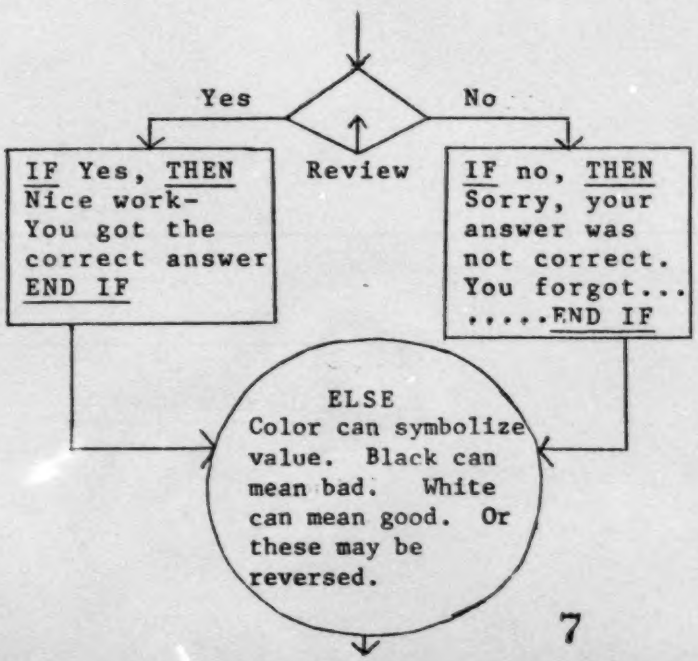


Thereafter, your program might be designed to invite the learner to make a choice about some thematic concept being in the passage. The invitation might, thus, be phrased "If this

excerpt uses color to symbolize the conflict between good and bad, press Y for YES. If not, press N for NO. If you wish to review, press R. This series of IF-THEN directives would similarly be housed in one rectangle, followed by a diamond.



After your learner has made his or her choice, your program must include "branches" that reinforce correct responses or remediate incorrect ones, and if you choose to review generally what you intended. Two of these will be IF-THEN statements, and one will be an ELSE statement. All three will be terminated with an END IF statement that generally reviews the intended concept about color symbolism, and is not a voluntary branch for the learner.



In a larger sense than this example shows, the limits of "Structured English" can be circumvented by adopting certain rhetorical guidelines offered by experience of humanistic CBI designs. These guidelines derive from a designer's understanding and empathizing with his or her audience to the extent that the computer program actually engages or leads the learner with the same kind of intuitive insight that a live teacher would use. Declarative sentences should be both free of jargon and brief. They should not patronize the learner when praising for completing a perfunctory or dumb step. So-called "Mand" sentences--commands, demands, and reprimands--should be made invitational and constructively supportive in phrasing. The designer should also construct real choices that make a difference, rather than perfunctory, empty interruptions. This means you could use instead of IF-THEN a rhetorical question with no expected answer, followed by an automatic branch to the general-concept review. Or the rhetorical question could be followed by several pseudo-choices, each of which is automatically explored in a follow-up branch, programmed as an END IF sequenced into DO-WHILE statements. Finally, in all verb phrases, there should be an almost Victorian sense of tentativeness, "You seemed to have pushed the key for a meaningless choice here. Want to try again?" Such language features help create the "User-Friendly" tone of good interactive computer programs and must, therefore, be criteria in the use of "Structured English."

Authoring systems, as mentioned earlier, relieve much of the need for planning with modified flowcharts. On the other

hand, the Information and Question pages within the authoring system will require knowledge and use of structured English. The following figures illustrate how the video screen displays the PASS authoring system.

INSERT "PASS" PAGES HERE

The first illustration shows the "menu," a list of options from which the user chooses a step in the pre-sequenced program. The menu obviously precludes the need for the user to draw the flowchart. Yet a flowchart can still be usable for demonstrating a kind of roadmap for the program as controlled by this authoring system. The arrow on this first illustration simulates a computer-display cursor, which is a moveable place mark controlled by the user.

The second illustration provides a look at the sub-components of "Lesson Development" on the original menu. Assuming the user wishes to view "Instructional Unit Editor" as shown, the screen would then display it as still another list of optional inputs from the user. Following the choice "Edit with Prompts" as shown, the user would then see the final menu list, consisting of an "Information Part" stimulus, the cue "Question Part," and the the follow-up branches to a "Response Part" and "Review Part." These four options really constitute the heart of any instruction's interaction and in fact, relate this part of the authoring system to the "Structured English" statements discussed earlier. The final screen-simulations show the blank spaces onto which the users typed information will display.

Note also that the user still may control his design here by requesting "no" page or by opting out entirely through a "Control Q" command or by editing pages through a "Replace" command. To further enhance computer displays, authoring systems offer various letter signs, colors, and graphic designs. Thus, by following the authoring system's reduction from abstract design to a more concrete question-response-reinforcement sequence, an English teacher can be like a chef following a recipe--adding only specific words as a seasoning to the whole experience.

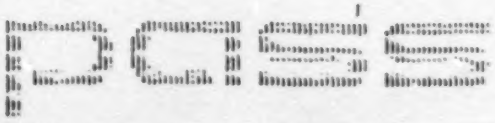
Though Structured English, modified flowcharts and authoring systems simplify the planning and delivery of computer based instruction, an English teacher still must be able to imagine what will display on the video monitor each step of the sequence. For this purpose, the "script page" is an excellent guide. Not unlike a film-makers shooting script, computer script pages contained in a loose-leaf notebook provide simply a precise communication between designer and programmer, so that both minds work as one, and the computer display will be exactly as conceived.

Note on the illustration how much information can be included on one script page. Predominant is the simulated video-screen grid, an exact facsimile of a 13" monitor screen, each block plotted by numbers so the programmer can convert whatever is in that block to a command that will eventually result in a display of letter or color in exactly that spot. Similarly important are the "branching" instructions, midpage left, that tell the programmer from which and to which displays this present one has come and will go, respectively. This information, together with the page number

in the upper right, is critical to keeping the program in order. Optional information may be added by the designer--including video disc sources of still or motion, computer audio like digital music or sound effects, or videodisc audio on either of two stereo tracks.

If all of this sounds very technical and heady, that's because it is. Planning computer-based instruction demands the most disciplined attention to every detail that one intends to take place, not unlike Poe's "ratiociration" or Sherlock Holmes' uncanny reconstructive imagination. But as teachers of literature or writing, English teachers have a reputation for being concerned with detail and concreteness. Thus, it is not alien, in concept, for our breed to reapply this concern to cybernetic thinking. Though rigorous, it is good mental exercise to go through the planning of a computer-based lesson. The left hemisphere immerses itself in sequential logic and detail, while the right envisions color, sound and graphics to engage the learner by ear and eye. Naturally, you won't be great right away; it's a trial-and-error process. But with a knowledge of the planning elements presented here, English teachers can assume their rightful place in the use of microcomputer instruction for worthwhile purposes.

Our profession can neither afford to pass up nor to be passed up by this medium. Fifteen years ago, John Culkin said "a lot has happened in the last twenty years and most of it plugs into the wall." Today, we can add the words of Chris Morgan, editor of BYTE magazine, "Never trust a computer you can't lift." Using the microcomputer effectively is as important for the future as learning to read. Thus, it falls within our area of professional expertise to know how to use it.



Professional Authoring Software System

→ 1. Lesson Development
2. Imagery Development
3. Learner Management
4. System Environment

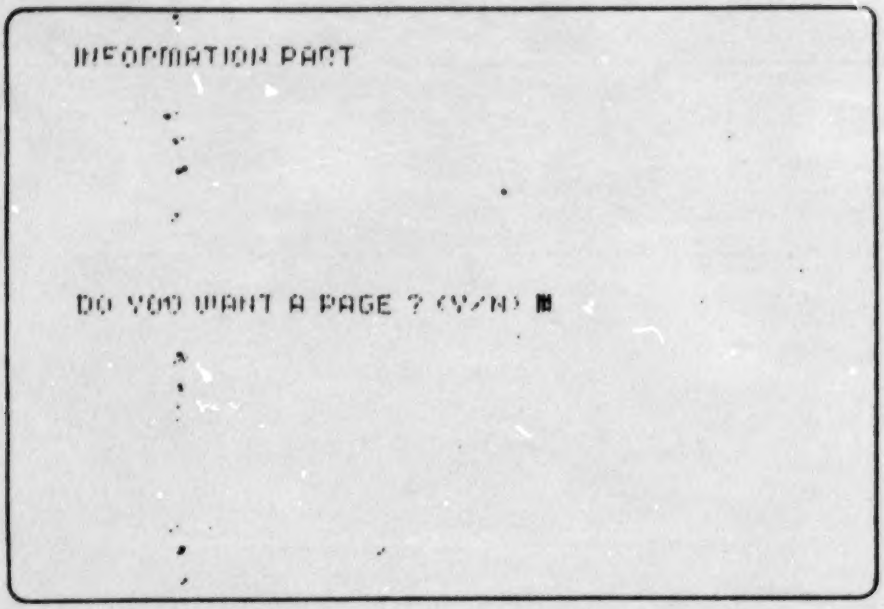
Enter Number
Control Q to QUIT

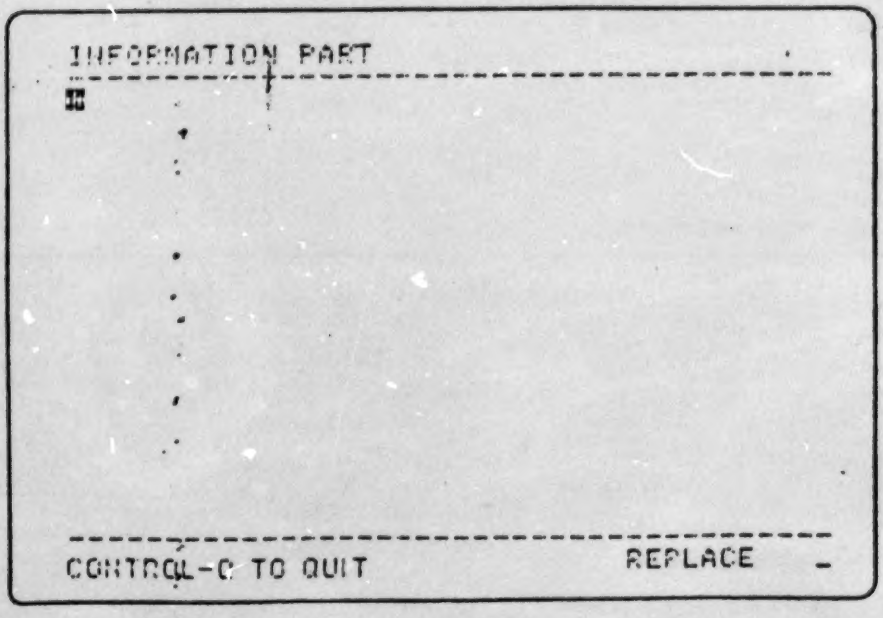
Lesson Development

1. Catalog
→ 2. Instructional Unit Editor
3. Branching Editor
4. List Lesson
5. Test Run Lesson

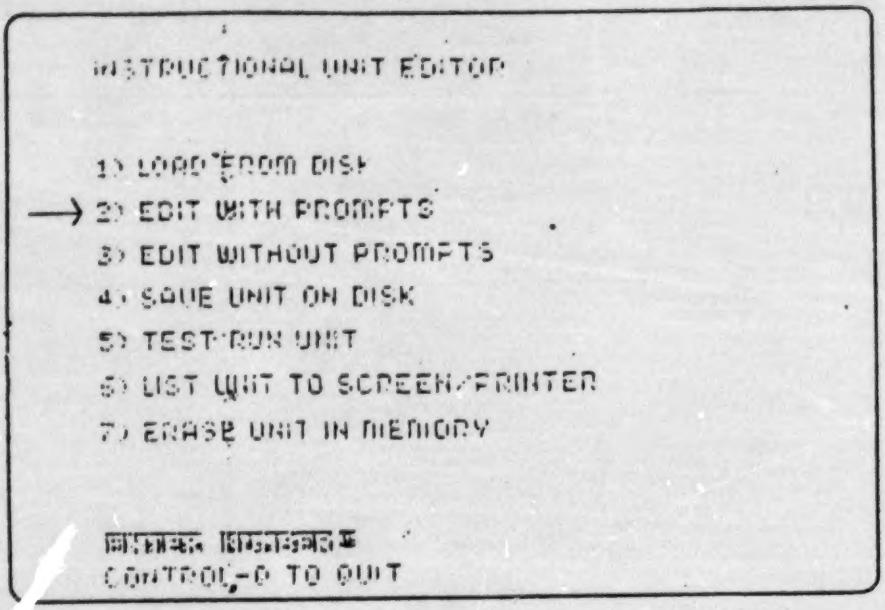
Enter Number
Control Q to QUIT

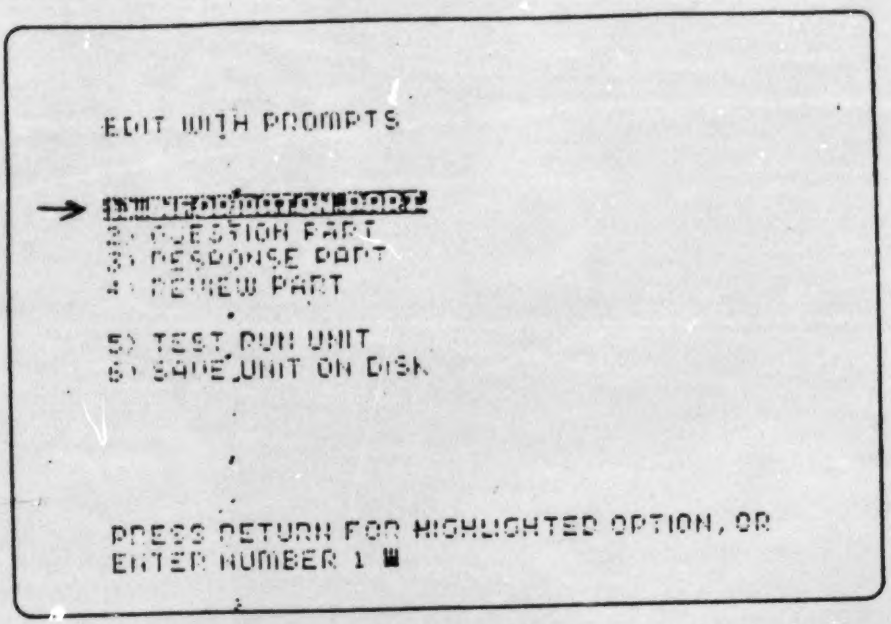
© 1981 Bell & Howell Company





© 1981 Bell & Howell Company


INSTRUCTIONAL UNIT EDITOR
1) LOAD FROM DISK
2) EDIT WITH PROMPTS
3) EDIT WITHOUT PROMPTS
4) SAVE UNIT ON DISK
5) TEST RUN UNIT
6) LIST UNIT TO SCREEN/PRINTER
7) ERASE UNIT IN MEMORY
CONTROL-C TO QUIT


EDIT WITH PROMPTS
2) QUESTION PART
3) RESPONSE PART
4) RENEW PART
5) TEST RUN UNIT
6) SAVE UNIT ON DISK
PRESS RETURN FOR HIGHLIGHTED OPTION, OR
ENTER NUMBER 1

VIDEODISC

Motion _____ Still _____ Time _____

Frame _____ to _____ Time Code _____

Text over still _____

pix in box _____ pale background _____

Other:

BRANCHING FROM:

BRANCHING GO TO:

COMPUTER AUDIO

COMPUTER

TEXTPORTS: A B 1 2 3 4 5 6

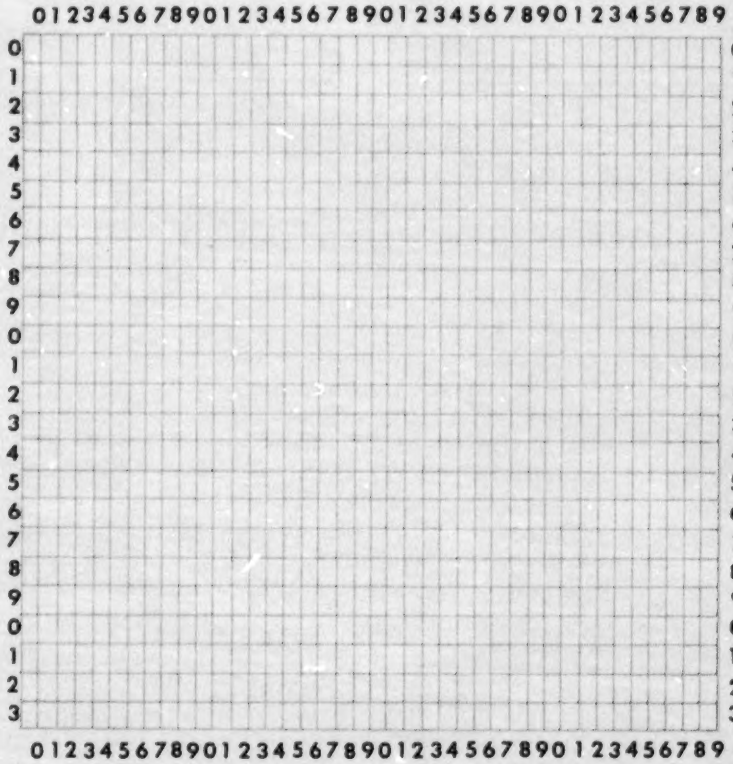
GRAPHICS:

PAGE

SEGMENT

MODULE

VISUAL



AUDIO 1 2